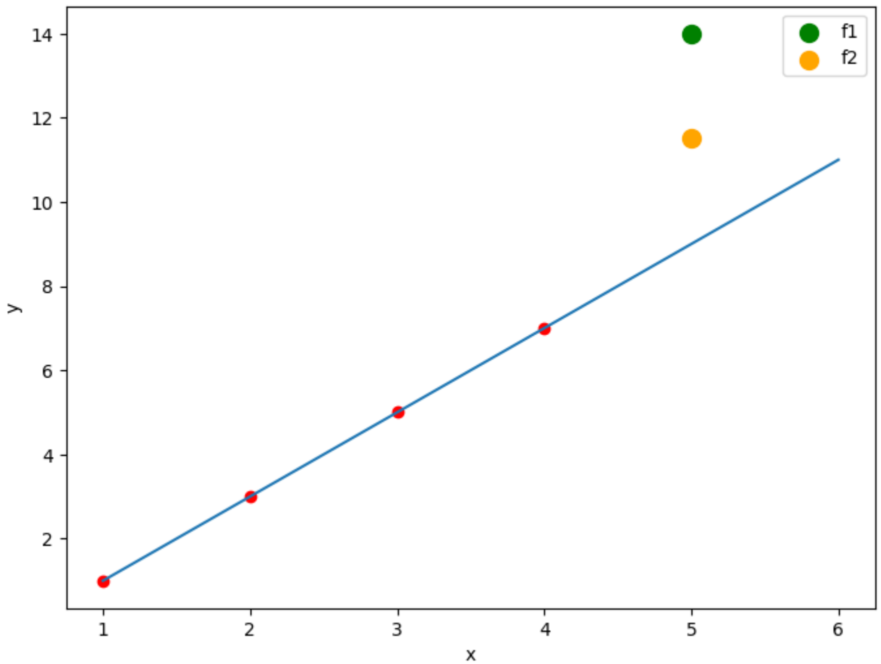
Gaussian Process Regression (GPR)
Finding distribution of functions

200677 Jun Sung Kim

Data points: (1,1), (2,3), (3,5), (4,7)
f(5) = ?



$$f_1(x) = \frac{91}{24}x^4 - \frac{455}{12}x^3 + \frac{3185}{24}x^2 - \frac{2251}{12}x + 90$$

$$f_1(5) = 100$$

$$f_2(x) = \frac{41}{24}x^4 - \frac{205}{12}x^3 + \frac{1435}{24}x^2 - \frac{1001}{12}x + 40$$

$$f_2(5) = 50$$

Our intuition : $f(5) = 9$

There is no perfect prediction function

- Instead, we can say the point we are finding for is probably somewhere.
- Given Data $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$
- Traditional method : find $y^* = f(x^*)$
- Now : find $P(y^*|D)$ (y^* is random variable)
- Do this for all $x^* \in I$: distribution of functions

- By Bayes' theorem ,

$$P(y^*|D) = \frac{P(D, y^*)}{P(D)} = \frac{P(y_1, \dots, y_n, y^*)}{P(y_1, \dots, y_n)}$$

(y_i is a random variable that indicates where the function value of x_i is likely to be.)

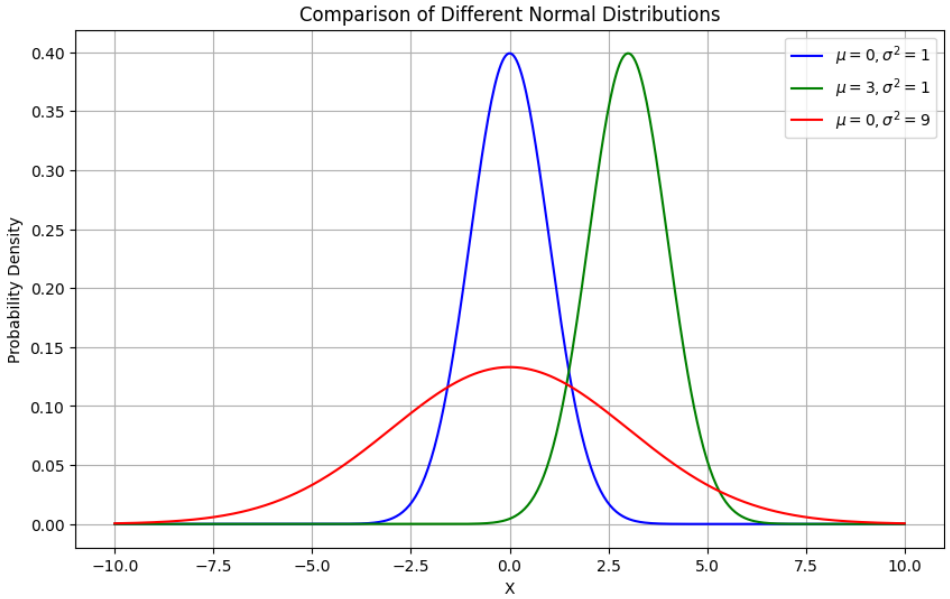
- We assume Gaussian Process :

  For all finite random variable $z_1, \dots, z_k$,

  $z_1, \dots, z_k \sim N(\mu, \Sigma)$ (Joint Gaussian Distribution)

Gaussian (Normal) Distribution

- $X \sim N(\mu, \sigma^2)$, $P(X = x) = \frac{1}{\sigma\sqrt{2\pi}} \mathrm{e}^{-\frac{(x-\mu)^2}{2\sigma^2}}$

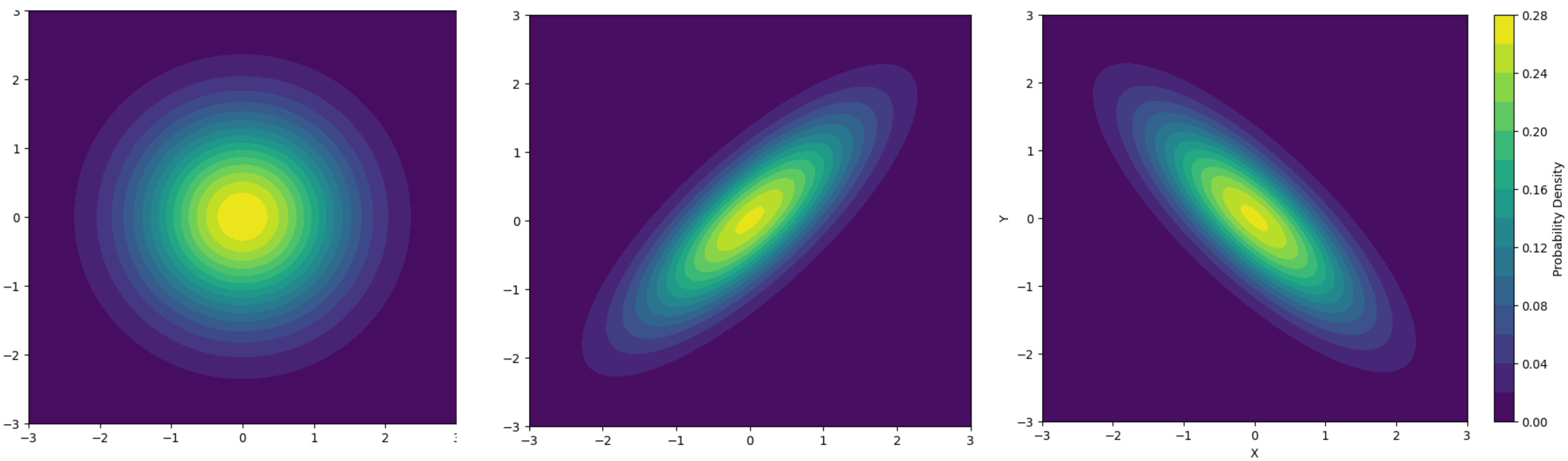


Joint (Multivariate) Gaussian Distribution

- X : n–dimensional random variable vector.
- $X \sim N(\mu, \Sigma)$
- Σ : n x n covariance matrix
- $P(X = \mathbf{x}\ ) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} \mathrm{e}^{-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)}$

2D Gaussian Distribution

$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 1 & -0.8 \\ -0.8 & 1 \end{bmatrix}$$



RBF Kernel Function

- $y_1, \ldots, y_n \sim N(\mu_1, \Sigma_1), y_1, \ldots, y_n, y^* \sim N(\mu_2, \Sigma_2)$
- We need to model μ_1, Σ_1, μ_2, Σ_2. $\mu_1 = \mu_2 = 0$
- $\Sigma_{(i,j)} = k\left(x_i, x_j\right) \coloneqq Ce^{-\frac{1}{2l^2}(x_i - x_j)^2}$
- C and l is hyperparameter for fitting data. (Optimization for $P(y_1, \ldots, y_n)$)

- $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\} = \mathbb{X} \times \mathbb{Y}$
- $\Sigma_1 = \begin{pmatrix} k(x_1, x_1) & \cdots & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \cdots & k(x_n, x_n) \end{pmatrix}$
- $\Sigma_2 = \left(\begin{array}{ccc|c} k(x_1, x_1) & \cdots & k(x_1, x_n) & k(x_1, x^*) \\ \vdots & \ddots & \vdots & \vdots \\ k(x_n, x_1) & \cdots & k(x_n, x_n) & k(x_n, x^*) \\ \hline k(x^*, x_1) & \cdots & k(x^*, x_1) & k(x^*, x^*) \end{array}\right) = \begin{pmatrix} \Sigma_1 & k(\mathbb{X}, x^*) \\ k(x^*, \mathbb{X}) & k(x^*, x^*) \end{pmatrix}$

$$P(y^*|D) = \frac{P(D, y^*)}{P(D)} = \frac{P(y_1, \dots, y_n, y^*)}{P(y_1, \dots, y_n)}$$

$$\begin{aligned}(y^*|D) \sim N(&k(x^*, \mathbb{X})\Sigma_1^{-1}\mathbb{Y}\,, \\ &k(x^*, x^*) - k(x^*, \mathbb{X})\Sigma_1^{-1}k(\mathbb{X}, x^*))\end{aligned}$$

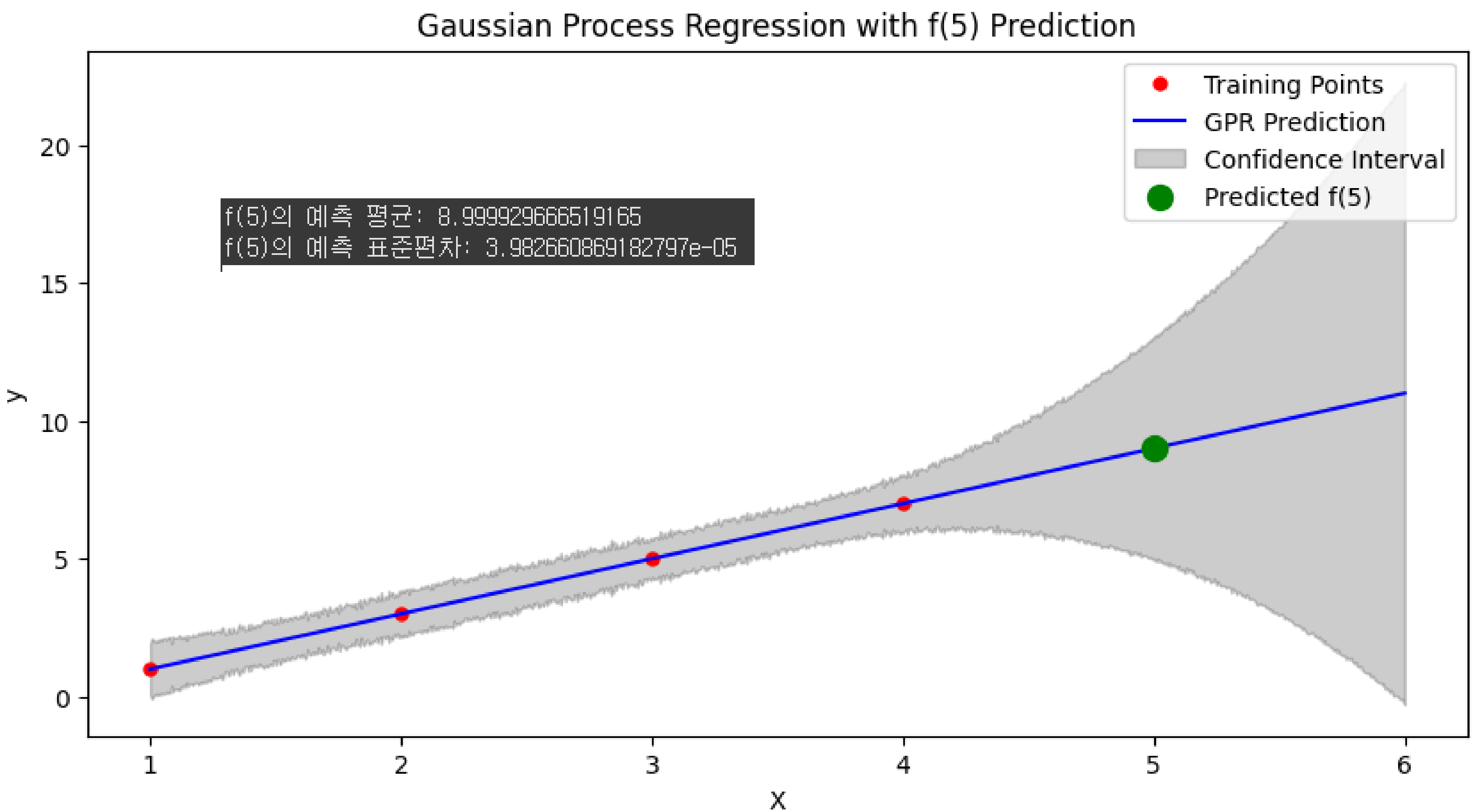
$\mathrm{E}(y^*|D)$: expected value

$\mathrm{V}(y^*|D)$: Confidence (신뢰도)

Example.

```python
# 주어진 데이터
X = np.array([[1], [2], [3], [4]])
y = np.array([1, 3, 5, 7])

# 커널 설정 및 GPR 모델 생성
kernel = C(1.0) * RBF(length_scale=1.0)
gp = GaussianProcessRegressor(kernel=kernel, n_restarts_optimizer=3)

# 모델(커널) 학습(최적화)
gp.fit(X, y)

# f(5)에 대한 예측 및 표준편차 계산
X_pred_single = np.array([[5]])
y_pred_single, sigma_single = gp.predict(X_pred_single, return_std=True)

# 결과 출력
print(f"f(5)의 예측 평균: {y_pred_single[0]}")
print(f"f(5)의 예측 표준편차: {sigma_single[0]}")

# 예측을 위한 새로운 X 범위 설정
X_pred = np.linspace(1, 6, 1000).reshape(-1, 1)
y_pred, sigma = gp.predict(X_pred, return_std=True)

# 결과 시각화
plt.figure(figsize=(10, 5))
plt.plot(X, y, 'r.', markersize=10, label="Training Points")
plt.plot(X_pred, y_pred, 'b-', label="GPR Prediction")
plt.fill_between(X_pred.ravel(), y_pred - 100000 * sigma, y_pred + 100000 * sigma, alpha=0.2, color='k', label="Confidence Interval")

# f(5) 예측값 시각화
plt.scatter(X_pred_single, y_pred_single, color='green', s=100, zorder=5, label="Predicted f(5)")

plt.xlabel("X")
plt.ylabel("y")
plt.title("Gaussian Process Regression with f(5) Prediction")
plt.legend()
plt.show()
```

Example.



Gaussian Process Regression with f(5) Prediction
Training Points
GPR Prediction
Confidence Interval
Predicted f(5)
f(5)의 예측 평균: 8.999929666519165
f(5)의 예측 표준편차: 3.982660869182797e-05
y
X
0
5
10
15
20
1
2
3
4
5
6

Pros and Cons

- Pros : You don't need to figure the model like polynomial, sine and cosine functions (i.e. , Non-parametric model, 비모수적 방법)
- Cons : Time complexity : $O(N^3)$

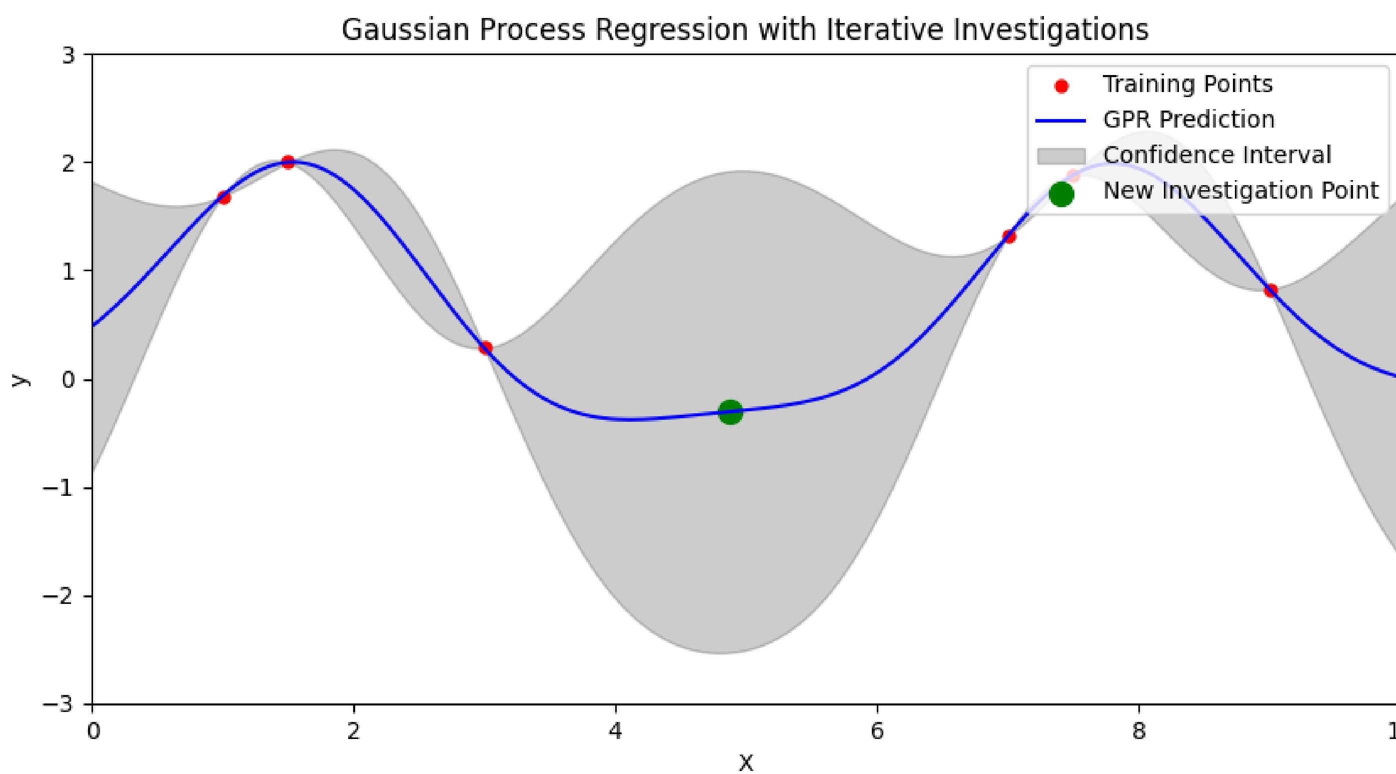
  Space complexity : $O(N^2)$

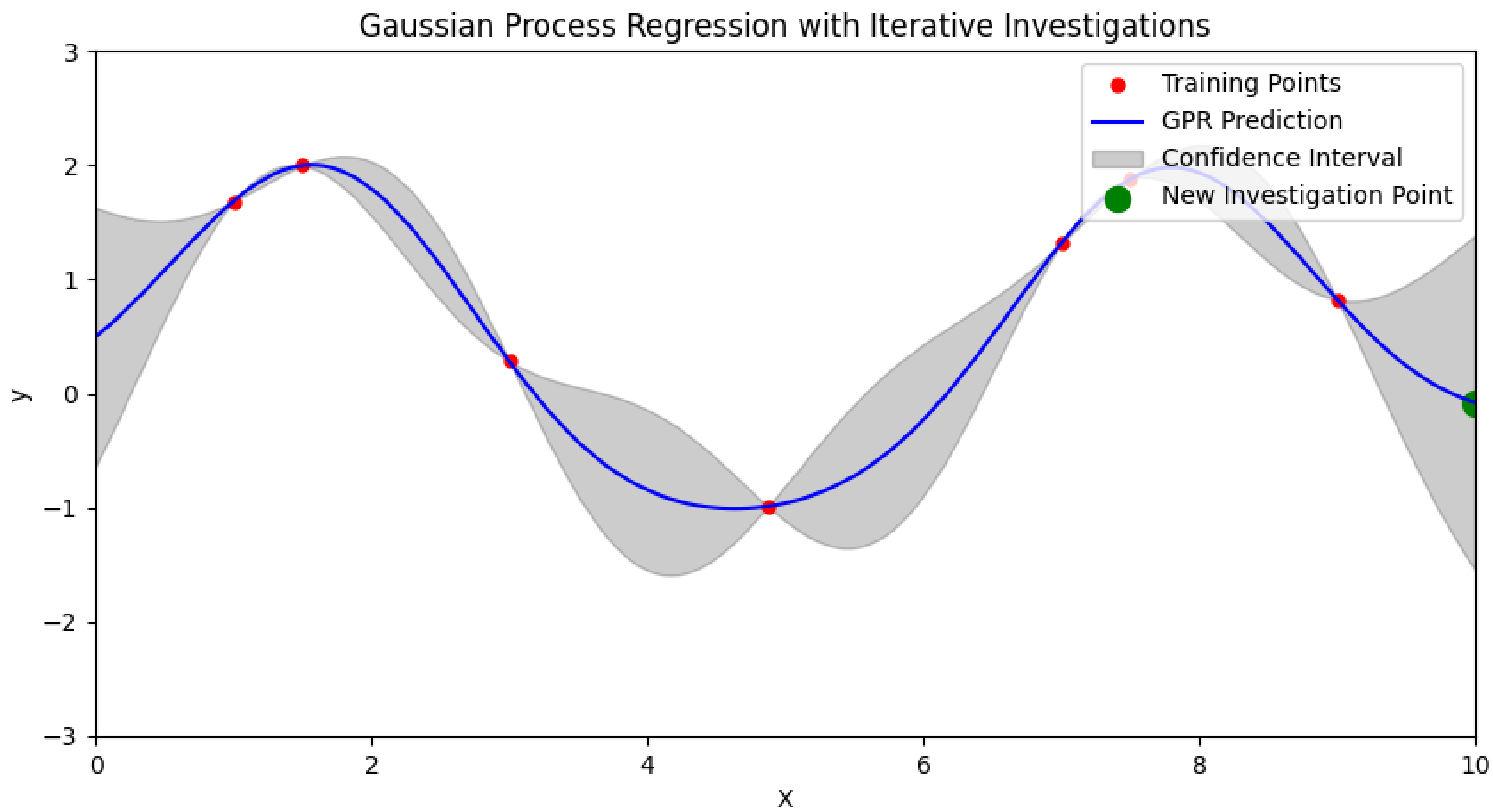
(N : Number of Data points)

Applications

Most efficient data to investigate (minimize error) (Related to Bayesian Optimization)

Select the point with the highest standard deviation





Gaussian Process Regression with Iterative Investigations
Training Points
GPR Prediction
Confidence Interval
New Investigation Point
3
2
1
0
−1
−2
−3
y
0
2
4
6
8
10
X

Bollinger Bands for stock chart

- m = the average price over 20 days
- 20 moving average (20MA, 20일 가격의 평균, 20일 이동평균선)

- σ = Standard deviation of the price over 20 days (20일 가격의 표준편차)

- Confidence Interval = [m-2σ,m+2σ] $\approx$ 95%

Central limit theorem

- $\frac{X_1+X_2+\cdots+X_N}{N} \sim N(\mu, \Sigma)\ as\ N \to \infty$
- $\frac{X_1+X_2+\cdots+X_{20}}{20}$ is close enough to $N(\mu, \Sigma)$.


Bitcoin / TetherUS PERPETUAL CONTRACT · 1날 · BINANCE
시97,960.5 고98,999.0 저96,929.9 종97,666.9 −293.5 (−0.30%)
97,716.0
셀
0.1
97,716.1
바이
3
USDT
50,000.0
49,000.0
48,000.0
47,000.0
46,000.0
45,000.0
44,000.0
43,000.0
40,720.9
40,668.8
40,595.4
40,446.9
40,373.1
40,298.9
40,000.0
39,000.0
38,093.7
37,000.0
36,000.0
35,000.0
34,000.0
33,000.0
32,000.0
31,000.0
30,000.0
29,000.0
TradingView